Real-Time Fall Detection Method Based on Hidden Markov Modelling

Alban Meffre, Christophe Collet, Nicolas Lachiche, and
Pierre Gangarski

University of Strasbourg-CNRS, LSIIT UMR 7005
Pôle API - Bd Sébastien Brant - BP 10413
67412 Illkirch CEDEX - France
{ameffre,c.collet,nicolas.lachiche,gancarski}@unistra.fr

Abstract. In the next few decades the increase of the number of elderly people will be of major concern, so that solutions must be found in order to maintain them at home. However such a population is exposed to the risk of falls, that can lead to dependency. This paper recalls some approaches used for fall detection and focuses on a method based on an uncalibrated camera. Motion detection uses a combination of simple Gaussian background modelling and interframe difference for person shape detection and features extraction. These features feed a Hidden Markov Model dedicated to fall detection. The algorithm has been tested on real data and we show that simple techniques can be used in order to obtain a fast and reliable fall detection system.

1 Introduction

1.1 Context

In the future the western European and American countries will have to meet the important challenge of elderly care. In 2050 the number of elderly people will rise from 9% to 15% in France, thanks to the stability of birthrate and the increase of life expectancy. Thus it will be necessary to increase the number of old people homes, and/or allow them to stay at home as long as possible.

Elderly people need to keep their partial autonomy in order to stay at home and falling is among the risks which can make them become dependent. After a fall elderly people often end up in hospital, and if not seriously injured, they lose their self-confidence and become more exposed to falling again. The aggravating factor with the fall is the time during which a person stays on the floor. The longer a person lies, the worse his situation becomes with a high risk of mortality.

Today a lot of devices exist for warning when elderly people fall. These devices can be automated and based on various technologies, such as accelerometers and inclinometers¹, or infrared motion sensors². Some devices are just remote alarm

¹ <http://www.fallsaver.net/>

² <http://www.pervaya.com/produit.html>

control and the person needs to manually activate the device after a fall, provided that he remains conscious, is able to move, and the device is properly worn.

Along with the expansion of information technologies and wireless communication systems, elderly people's care at home becomes an important subject for researchers and every improvement in this domain is a step to overcoming the problem of dependency. Over the past few years many studies have been carried out concerning fall detection by means of sensors integrated within the home.

In this way, the fall detection by video camera can be achieved by different means. Some methods are based on 3D reconstruction but need the use of one or several calibrated cameras. Nevertheless we will focus in this paper on 2D video image analysis with a single uncalibrated camera, for simplicity and cost reasons.

We present in the following different approaches related to single camera fall detection, whereas Sect. 2 presents video processing and detection with a Markovian model. Next we give the overview of the method we use and then we validate it on real data (Sect. 3). Finally we conclude with a discussion about the possible improvements of our work.

1.2 Related Work

The first step for fall detection consists in detecting the motion using a camera. The interframe difference is a straightforward method that consists in building a binary mask with the thresholded absolute difference between two subsequent images. It is used by authors such as Hsieh et al. [9] or Lee et al. [11] in combination with another background subtraction method. The simple Gaussian background subtraction method, used by Töreyin et al. [16] and Dahmane et al. [6], is another fast and straightforward method and usually sufficient for indoor scenes. Easy to implement, this method does not take into account a moving background. Some authors use the mixture of Gaussian by Stauffer [14] or the codebook background subtraction by Kim et al. [10]. These methods are more suitable for outdoor scenes. Hsieh et al. [9] combine image difference with a simple Gaussian background, and Lee et al. [11] combine it with a Gaussian mixture. Shadow suppression can be done in the HSV color space [4] but is time consuming. Shadow suppression methods in RGB color space are presented by Dahmane et al. [6] and Kim et al. [10]. Tracking can be implemented in order to follow several people in the scene. Lee et al. [11] use a variant of the Kalman filter. Tracking can be useful for occlusion handling but it requires complex models [5].

The second step is the feature extraction needed for fall detection. This is performed on the binary mask obtained with motion detection or background segmentation. The straightforward method consists in surrounding the moving shape (or blob) with a bounding box as in [17,1,11,7], and/or a fitted ellipse [12,7]. Thus the height to width ratio and inclination angle are retrieved. In

Rougier et al. [13] the 3D spatio-temporal trajectory of the head is used for detecting the fall. Sound can be used as well for detecting events [16] and Foroughi et al. [8] use projection histograms, along with Discrete Fourier Transforms (DFT). These methods require dimensionality reduction by Principal Component Analysis (PCA) for easier handling and computing. In certain cases some information about motion quantity is needed. The Motion History Image (MHI) and Motion Energy Image (MEI) were introduced by Bobick et al. [3].

The final decision taking of fall detection systems is made by a classification algorithm. Neural networks can be used [7], as well as Support Vector Machines [8]. These techniques need a supervised learning. Hidden Markov Models (HMM) are well suited for fall detection based on video. Anderson et al. [1] use one HMM per posture detection, and Töreyn et al. [16] detect falls from both audio and video.

2 Fast Fall Detection Method

The general principle of our algorithm (Fig. 1) is separated into two distinct parts. The first (Sect. 2.1) is based on a simple Gaussian background subtraction and interframe difference based motion detection, followed by a shape analysis and feature extraction with a fitted ellipse and a bounding box. In the second one (Sect. 2.2) the features are undersampled and classified with a Bayesian detection process, and the label sequence combined with ground truth for performance evaluation.

Fig. 1. Diagram of the video analysis

2.1 Video Processing and Feature Extraction

2.1.1 Initialization

Both the background model and interframe difference need to be initialized at program start. This is done without any movement in the scene, because only pixel noise has to be measured.

The background model initialization consists in accumulating the average and variance of input video images over time [6], in our work we choose the learning rate $\alpha = 0.95$ and thus initialization requires a few tens of images.

We assume that global illumination remains constant over time or changes very slowly. Cases with strong illumination changes are special cases that need to be processed separately and induce resetting the background model.

2.1.2 Background Subtraction and Mask Refining

The Gaussian background subtraction method results in a binary mask. The background maintenance is achieved by computing the temporal average and variance of the pixel intensities over the three RGB channels (1), (2). Let $\alpha \in]0; 1[$ represent the learning rate at which the input image I_n is added into the background model B_n . V_n can be considered as the variance of pixel s values in each RGB channel. The background model is updated recursively depending on the mask $\mathcal{UM}$.

$$B_n(s) = \begin{cases} \alpha \cdot B_{n-1}(s) + (1 - \alpha) \cdot I_n(s) & \text{if } s \notin \mathcal{UM} \\ B_{n-1}(s) & \text{if } s \in \mathcal{UM} \end{cases} \quad (1)$$

$$V_n(s) = \begin{cases} \alpha \cdot V_{n-1}(s) + (1 - \alpha) \cdot (B_n(s) - I_n(s))^2 & \text{if } s \notin \mathcal{UM} \\ V_{n-1}(s) & \text{if } s \in \mathcal{UM} \end{cases} \quad (2)$$

The segmentation is done by comparing each pixel s in the input image I_n with the background model B_n in the three RGB channels. If the difference is greater than 3 times the standard deviation $\sqrt{V_n(s)}$ for at least one of the RGB channels, the pixel is marked as foreground in the segmentation mask, and background otherwise. A pixel marked as foreground can correspond to a shadow. The shadow detection follows the method described in [6], wherein thresholds are respectively 0.44 and 1.09 for the lower and higher brightness distortion limits, and 7 for the color distortion.

Noise removal is performed by an erode-dilate morphological operation using a 3 by 3 square structural element and all the blobs that have an area smaller than 300 pixels are removed.

2.1.3 Feature Extraction

The features needed for fall detection are extracted in the mask (Fig.2). The fitted ellipse gives the angle α_n of the main axis of the blob, and the bounding box around the blob gives the height to width ratio ρ_n . The topmost point of the blob gives the displacement speed Vh_n of the head of the person. After the feature extraction the blob is replaced by its convex hull by redrawing the mask.

2.1.4 Motion Detection and Mask Refining

To detect the motion we use the binarized absolute image difference D_n . Let G be the gain and Ig_n the gray levels input image, so $D_n = (G \cdot |Ig_n - Ig_{n-1}|) \overset{255}{\underset{0}{\gtrless}}$

Fig. 2. Extraction of the features α_n and ρ_n

128. The interframe difference shows zones that have changed, i.e. only the outline of a moving object appears in the mask. The empty zones left are partially filled by expanding the outline with a dilate operation performed at quarter resolution by double down-sampling with Gaussian filtering. Then the mask is up-sampled twice toward the original resolution. The gain G is initialized in order to keep a ratio of 0.1% of white pixels when there is no movement in the scene.

2.1.5 MHI and MEI

The refined motion mask is used for updating the MHI (Bobick et al.[3]). The decay rate between two subsequent images is of 10 gray levels. The MEI is obtained by thresholding the MHI at gray level 1.

2.1.6 Motion Quantification and Background Maintenance

In Sect. 2.1.2, the background is updated with respect to the update mask $\mathcal{UM}$. When motion is present in the scene, the MEI is copied to the update mask $\mathcal{UM}$ and thus the background model “absorbs” slight changes in the scene like a previously displaced object. When no motion is present in the scene the background subtraction mask is copied to $\mathcal{UM}$, preventing the stationary person becoming integrated in the background model.

The motion is quantified by measuring the ratio m of the number of white pixels in MEI to the number of pixels in the background subtraction mask. The threshold between the two cases is set to 2.

2.2 Fall Detection

The fall detection is achieved in four steps and separately from the video processing. The whole feature data is processed at once after it has been extracted from the video sequences (Fig.3).

2.2.1 Pre-filtering of the Features

The video processing and feature extraction provide features at a rate of 25 datasets per second. Actually we do not need to detect falls at such a rate so the features are filtered and under-sampled before the detection. The angle α_n and height to width ratio ρ_n are averaged over 50 samples every 25 samples and hence we obtain α_t and ρ_t at a rate of 1 dataset per second. In order to

Fig. 3. Data pre-processing, classification and post-processing of states S_1 = "non-fall" and S_2 = "fall"

remove spurs the speed of the head Vh_n is first filtered by a Butterworth filter using the general formula $y_n = b_0x_n + b_1x_{n-1} - a_1y_{n-1}$ where $a_1 = -0.7265$ and $b_0 = b_1 = 0.13675$, then $\max(Vh_n)$ is kept over 50 samples every 25 samples so that we obtain Vh_t at a rate of one dataset per second.

2.2.2 Detection

In our work we assume that the postures of a person with time follows a Markov process. The observation is formed by the ρ_t and α_t information and the two hidden states (or classes) are labelled "fall" and "non-fall" with two parameters per class corresponding to the average and standard deviation of the normal distribution of the observation given the class. The labeling process is performed using the Baum-Welch procedure[2] and an Expectation Maximization (EM) algorithm[15] is used for the hyper-parameters estimation. These hyper-parameters are initialized with $\{\mu_\rho = 3; \sigma_\rho^2 = 1; \mu_\alpha = 90; \sigma_\alpha^2 = 10\}$ for the "non-fall" label and $\{\mu_\rho = 0; \sigma_\rho^2 = 1; \mu_\alpha = 0; \sigma_\alpha^2 = 100\}$ for the "fall" label, where μ and σ^2 are the average and variance of the features. Each iteration of the loop is performed as following. The forward probabilities are calculated using scaling. Then the backward probabilities are calculated and the product of forward and backward variables is maximized in order to retrieve the hidden states chain. This is the "Maximization" step. The Log-Likelihood of the data driven term is calculated by summing the scaling variables over the whole chain. The "Expectation" step consists in re-evaluating the parameters of the Markov model. First the gaussian parameters of the data driven term are evaluated given the new sequence of hidden states (i.e. the average values and covariances matrices), and second the transition probabilities and state initial probabilities are calculated.

The loop of the EM algorithm ends when the Log-Likelihood has converged to a steady value or when 100 iterations have been performed. Then the sequence of labels that maximize the likelihood of each observation is returned.

This procedure returns the maximum a posteriori hidden states sequence given the observation $\{\rho_t; \alpha_t\}$.

2.2.3 Data Post-Processing and Performance Evaluation

The classification takes into account only static labels such as “non-fall” and “fall”. Hence the speed of the head Vh_t and the duration of a posteriori occurrence of “fall” states are used after classification in order to remove false positives. A true positive is obtained when the duration of the occurrence of the “fall” state is greater than 5 seconds, and when the speed Vh_t is greater than 5 at the very beginning of the state occurrence.

As the video sequences are manually annotated, the result of the classification process is compared with the ground truth. This is done by combining them with a finite state machine, in order to prevent multiple detection of a fall during an actual “fall” state. So the real number of actual falls in video sequences is estimated.

3 Experimental Results

The video processing and feature extraction part of the program is written in C++ using the *OpenCV2.2* library and the classification and fall detection is performed using *Matlab*. The method has been tested on a set of 24 video clips available at <http://lsiit-miv.u-strasbg.fr/lsiit/perso/collet/Images/videosENSPS/>. The videos have a standard format of 640×480 pixels at a framerate of 25 images per second. We obtained a 34 images per second process rate on an *intel core i5 750* processor. Different kinds of situations have been simulated with the help of actors, such as walking, falling with different angles, bending down, sitting on a chair, carrying an object, etc. These actions are combined with lighting variations and partial occlusions.

We performed the classification on the whole video set and we obtained 90% sensitivity and 100% specificity. Some falls have not been detected because our system is not able to deal with complete occlusion. If we do not take into account the speed of the head Vh_t , the specificity drops to 96%. Therefore the speed of the head helps to eliminate false positives.

The Fig.4 gives examples of falls properly detected. Even if the fall occurs in the camera axis, the fall is detected as long as the head speed is high enough. The fall will be detected too if the fall occurs in another direction and the body is masked by small object only, so the angle of the fitted ellipse will correspond to the angle of the entire body. Small object displaced in the scene will do not affect the behavior of the background segmentation algorithm because they do not move once put down, so they are quickly absorbed in the background model. However strong illumination changes or hard shadows can lead to false positives or false negatives because in these cases the fitted ellipse surrounds both the body of the person and the undesirable artifacts.

An example of fall positive is given with Fig.5 where the person fall behind a big object. As the body is partially or totally hidden, the ellipse fitting can not perform properly.

Fig. 4. Examples of true positives and true negatives: falling in the camera axis(a), falling with slight occlusion(b) carrying an object(c)

Fig. 5. Example of false negative: falling with strong occlusion

4 Discussion and Conclusion

After we read the different papers related to our work, we retain that the complexity of the methods varies a lot depending on which goal the authors expected to reach. If we want to follow several people in the scene or deal with strong occlusions, hard shadows and strong illumination changes, we have to implement tracking and/or pattern recognition algorithms. These techniques tend to use a lot of computing and memory resources.

In our case the aim was to verify that we can implement a reliable fall detection system with robust and fast techniques and thus we use the simple background modelling and motion detection, in combination with a Bayesian classification. Therefore we show that combining them wisely gives good classification results and a high video processing rate. In the meantime a video system cannot be used alone for reliable care of elderly people. Good performance and security can be achieved by a use of our algorithm in combination with other external sensors integrated into the home environment such as Passive-InfraRed (PIR) detectors.

We would like to thank Region Alsace for funding and partial support of this work.

References

1. Anderson, D., Keller, J.M., Skubic, M., Chen, X., He, Z.: Recognizing falls from silhouettes. In: 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2006, pp. 6388–6391 (2006)
2. Baum, L.E.: An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities* 3, 1–8 (1972)
3. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(3), 257–267 (2001)
4. Cucchiara, R., Grana, C., Piccardi, M., Prati, A., Sirotti, S.: Improving shadow suppression in moving object detection with HSV color information. In: Proc. IEEE Intl. Conf. Intelligent Transportation Systems, pp. 334–339 (2001)
5. Cucchiara, R., Vezzani, R.: Assessing temporal coherence for posture classification with large occlusions. In: IEEE Workshop on Motion and Video Computing, WACV/MOTIONS 2005, vol. 2, pp. 269–274 (2005)
6. Dahmane, M., Meunier, J.: Real-time moving object detection and shadow removing in video surveillance. In: 3rd International Conference of SETIT, Tunisia (2005)
7. Froughi, H., Aski, B.S., Pourreza, H.: Intelligent video surveillance for monitoring fall detection of elderly in home environments. In: 11th International Conference on Computer and Information Technology, ICCIT 2008, pp. 219–224 (2008)
8. Froughi, H., Rezvani, A., Pazirae, A.: Robust fall detection using human shape and multi-class support vector machine. In: Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, pp. 413–420 (2008)
9. Hsieh, C.-C., Hsu, S.-S.: A simple and fast surveillance system for human tracking and behavior analysis. In: Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, SITIS 2007, pp. 812–818 (2007)

10. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* 11(3), 172–185 (2005)
11. Lee, Y.-S., Lee, H.J.: Multiple object tracking for fall detection in real-time surveillance system. In: 11th International Conference on Advanced Communication Technology, ICACT 2009, vol. 03, pp. 2308–2312 (2009)
12. Nait-Charif, H., McKenna, S.J.: Activity summarisation and fall detection in a supportive home environment. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 4, pp. 323–326 (2004)
13. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Monocular 3d head tracking to detect falls of elderly people. In: 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2006, pp. 6384–6387 (2006)
14. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 252 (1999)
15. Tanner, M.A.: Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions. Springer (1996)
16. Ugur Toreyin, B., Dedeoglu, Y., Enis Cetin, A.: HMM based falling person detection using both audio and video. *Computer Vision in Human-Computer Interaction*, 211–220 (2005)
17. Vishwakarma, V., Mandal, C., Sural, S.: Automatic Detection of Human Fall in Video. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) *PREMI 2007*. LNCS, vol. 4815, pp. 616–623. Springer, Heidelberg (2007)